

593-61
ABS ONLY
175083
N 94-22531

A NEW PROGRAMMING METAPHOR FOR IMAGE PROCESSING PROCEDURES

O.M.Smirnov (*Inst. of Astr. of the Russian Ac. Sci.*)
N.E.Piskunov (*Obs. and Astrophys. Lab., U. of Helsinki*)

Most image processing systems, besides an Application Program Interface (API) which lets users write their own image processing programs, also feature a higher level of programmability. Traditionally, this is a command or macro language, which can be used to build large procedures (scripts) out of simple programs or commands. This approach, a legacy of the teletypewriter, has serious drawbacks. A command language is clumsy when (and if!) it attempts to utilize the capabilities of a multitasking or multiprocessor environment, it is but adequate for real-time data acquisition and processing, it has a fairly steep learning curve, and the user interface is very inefficient, especially when compared to a graphical user interface (GUI) that systems running under X11 or Windows should otherwise be able to provide. All these difficulties stem from one basic problem: a command language is not a natural metaphor for an image processing procedure.

The paper describes in detail a more natural metaphor—an *image processing factory*. A factory is a set of programs (applications) that execute separate operations on images, connected by *pipes* that carry data (images and parameters) between them. The programs function concurrently, processing images as they arrive along pipes, and querying the user for whatever other input they need. From the user's point of view, programming (constructing) factories is a lot like playing with LEGO blocks—much more intuitive than writing scripts.

The paper focuses on some of the difficulties of implementing factory support, most notably the design of an appropriate API. It also shows that factories retain all the functionality of a command language (including loops and conditional branches), while suffering from none of the drawbacks outlined above. Other benefits of factory programming include self-tuning factories and the process of encapsulation, which lets a factory take the shape of a standard application both from the system and the user's point of view, and thus be used as a component of other factories.

A bare-bones prototype of factory programming has been implemented under the **PCIPS** image processing system, and a complete version (on a multitasking platform) is under development.